SozKZ: Training Efficient Small Language Models for Kazakh from Scratch

Saken Tukenov
Independent Researcher
saken@tukenov.kz

Kazakh, a Turkic language spoken by over 22 million people, remains underserved by existing multilingual language models, which allocate minimal capacity to low-resource languages and employ tokenizers ill-suited to agglutinative morphology. We present SozKZ, a family of Llama-architecture language models (50M–600M parameters) trained entirely from scratch on 9 billion tokens of Kazakh text with a dedicated 50K BPE tokenizer. We evaluate all models on three Kazakh benchmarks—multiple-choice cultural QA, reading comprehension (Belebele), and topic classification (SIB-200)—alongside five multilingual baselines ranging from 500M to 3B parameters. Our 600M model achieves 30.3% accuracy on Kazakh cultural QA, approaching the 32.0% of Llama-3.2-1B (2× larger), and 25.5% on SIB-200 topic classification, surpassing all evaluated multilingual models up to 2B parameters. We observe consistent scaling from 50M to 600M, with MC QA accuracy rising from 22.8% to 30.3%, suggesting that further scaling remains beneficial. These results demonstrate that small, dedicated models trained from scratch with a language-appropriate tokenizer offer a viable path for low-resource language technology, achieving competitive performance at a fraction of the computational cost. All models and the tokenizer are released under open licenses.

1 Introduction

Kazakh is a Turkic language spoken by approximately 22 million people, primarily in Kazakhstan and neighboring countries. As an agglutinative language with rich morphology, Kazakh poses particular challenges for natural language processing: words carry extensive inflectional and derivational suffixes, leading to a large effective vocabulary that general-purpose multilingual tokenizers handle poorly. Despite a growing body of NLP research for Kazakh [25, 26], the language remains underserved by the current generation of large language models, which typically allocate only a fraction of their capacity to low-resource languages.

Modern multilingual models such as Llama 3 [23], Gemma [9], Qwen 2.5 [4], and Mistral [11] are trained on corpora dominated by English and other high-resource languages. When applied to Kazakh, these models face two compounding inefficiencies. First, their tokenizers fragment Kazakh text into disproportionately many subword tokens—a phenomenon quantified by *fertility*, the average number of tokens per word. Second, the vast majority of model parameters encode knowledge about languages other than Kazakh, yielding a poor ratio of useful capacity to total model size.

We argue that **small, dedicated models can achieve competitive performance with much larger multilingual models on Kazakh language tasks at a fraction of the computational cost**. To test this thesis, we train SozKZ, a family of Llama-architecture causal language models at four parameter scales—50M, 150M, 300M, and 600M—entirely from scratch on a curated Kazakh corpus of approximately 9 billion tokens.

Our contributions are as follows:

1. We present an end-to-end pipeline for training Kazakh language models from scratch at four scales (50M–600M parameters), including data collection, cleaning, and pre-tokenization.
2. We design a dedicated ByteLevel BPE tokenizer with a 50K vocabulary[1] that achieves a 2–3× fertility advantage over multilingual tokenizers on Kazakh text.
3. We evaluate SozKZ models against five multilingual baselines (Qwen, Llama 3, Gemma) spanning 0.5B–3B parameters on three Kazakh benchmarks, showing competitive performance and a clear advantage on topic classification.
4. We report empirical scaling curves for Kazakh language modeling, identifying the efficiency sweet spot where dedicated small models offer the best performance-per-parameter trade-off.
5. We release all models, the tokenizer, and the training pipeline under open licenses to support further research on Kazakh NLP.

The rest of this paper is organized as follows. Section 2 surveys related work on Kazakh NLP, low-resource language modeling, scaling laws, and tokenizer design. Section 3 describes our data pipeline, tokenizer, model architectures, and training procedure. Section 4 details the benchmark suite and evaluation protocol. Section 5 presents and analyzes results. Section 6 summarizes our findings and outlines future directions.

2 Related Work

2.1 Kazakh NLP

Research on Kazakh natural language processing has accelerated in recent years, driven largely by the IS2AI research group at Nazarbayev University. Yeshpanov et al. [25] introduced KazNERD, a large-scale named entity recognition dataset for Kazakh covering 25 entity classes across news, fiction, and legal domains. Yeshpanov et al. [26] subsequently released KazQAD, a question-answering dataset for Kazakh built from Wikipedia and educational texts. These resources have enabled systematic evaluation of NLP models on Kazakh for the first time.

Beyond datasets, several efforts have targeted Kazakh language modeling directly. Multilingual models such as mBERT [8] and XLM-R [7] include Kazakh in their training data but allocate minimal capacity to it. More recently, KazLLM [28] explored continued pre-training of large models on Kazakh corpora. Our work differs in training dedicated models entirely from scratch, allowing full control over architecture, tokenizer, and data composition.

2.2 Low-Resource Language Models

Training language models for low-resource languages has emerged as an active research area. Ogueji et al. [16] demonstrated with AfriBERTa that small transformer models trained on modest corpora (less than 1 GB) can achieve competitive performance on African languages, challenging the assumption that large-scale data is a prerequisite. For Arabic, Sengupta et al. [18] trained Jais, a 13B-parameter model from scratch on a bilingual Arabic-English corpus, showing that dedicated models outperform multilingual ones on Arabic-specific benchmarks.

[1] `https://huggingface.co/stukenov/kazakh-bpe-32k`

SEA-LION [3] adopted a similar philosophy for Southeast Asian languages, training models at multiple scales with language-specific tokenizers. The BLOOM project [6] took a different approach, training a single 176B-parameter multilingual model on 46 languages with the goal of equitable coverage. Our work follows the dedicated-model approach of Jais and AfriBERTa but at substantially smaller scales (50M–600M parameters), testing how far efficiency gains from specialization can compensate for reduced model capacity.

2.3 Scaling Laws

Kaplan et al. [12] established power-law relationships between model size, dataset size, compute budget, and cross-entropy loss for neural language models. Hoffmann et al. [10] refined these findings with the Chinchilla scaling laws, demonstrating that many large models are significantly undertrained relative to their parameter count and that a compute-optimal model should be trained on approximately 20 tokens per parameter.

In practice, the Llama model family [23, 24] demonstrated that training smaller models on substantially more data than the Chinchilla-optimal ratio yields models that are more efficient at inference time. Our training regime follows this over-training philosophy: the SozKZ-600M model is trained on approximately 9B tokens, yielding a tokens-to-parameters ratio of roughly 15:1—slightly below the Chinchilla optimum but reflective of the available Kazakh data.

These scaling laws have been derived primarily from English-language experiments. Whether they transfer to morphologically rich, low-resource languages with different tokenizer efficiencies remains an open question that our scaling analysis addresses.

2.4 Tokenizer Design

Subword tokenization, introduced by Sennrich et al. [19] with Byte-Pair Encoding (BPE) and refined by Kudo and Richardson [13] with SentencePiece, is now standard in language model pre-training. The choice of tokenizer has outsized effects on model efficiency for morphologically rich languages [17]: a tokenizer trained on English-dominated data will fragment agglutinative languages like Kazakh, Turkish, or Finnish into far more tokens per word than necessary, effectively reducing the model's context window and increasing inference cost per semantic unit.

Fertility—the average number of tokens per whitespace-separated word—is a widely used metric for quantifying this inefficiency [1]. Prior work on Turkic languages has shown that dedicated tokenizers achieve fertility values 2–3× lower than multilingual alternatives [22].

We train a ByteLevel BPE tokenizer with a 50K vocabulary exclusively on Kazakh text and quantify its fertility advantage over the tokenizers used by Llama 3, Gemma, Qwen 2.5, and Mistral. This dedicated tokenizer is a key component of the efficiency gains we observe, as lower fertility translates directly to more content per fixed context window.

3 Methodology

This section describes the full training pipeline for the SozKZ model family: data collection and cleaning, tokenizer design, model architecture, and training configuration. All code, configs, and trained models are released publicly to enable full reproducibility.

3.1 Training Data

We construct a large-scale monolingual Kazakh corpus by aggregating text from 18 web and curated sources, including CulturaX [7], HPLT 2.0, mC4 [8], MADLAD-400, mOSCAR, Kazakh Wikipedia, and the `kz-transformers/multidomain-kazakh-dataset`. The raw collection comprises 28.4M documents.

We apply a 9-stage cleaning pipeline:

1. Unicode NFC normalization,
2. removal of control characters,
3. whitespace and newline collapsing,
4. minimum length filtering (≥ 50 characters),
5. URL density filtering (≤ 5 per 1,000 characters),
6. HTML tag filtering (≤ 5 tags),
7. Kazakh character ratio filter (script-based),
8. language identification (retaining only Kazakh),
9. exact deduplication via MD5 hashing (both within-source and cross-source against the existing `multidomain-kazakh-dataset` reference of 12.4M unique hashes).

After cleaning, 13.7M documents remain (48.2% pass rate), yielding approximately 9.0B tokens under our BPE 50K tokenizer. This constitutes one of the largest publicly available monolingual Kazakh corpora. We note the inherent data-quality versus quantity tradeoff in low-resource settings: aggressive filtering removes noise but discards potentially useful text, while lenient filtering risks training on low-quality data. Our 48.2% pass rate reflects a moderately conservative stance.

3.2 Tokenizer

We train a ByteLevel BPE tokenizer [19] with a vocabulary of 50,257 tokens exclusively on Kazakh text from our cleaned corpus.

The motivation for a dedicated tokenizer is twofold. First, multilingual tokenizers (e.g., those used by Llama, Qwen, or Gemma) allocate vocabulary capacity across 100+ languages, resulting in poor *fertility*—the average number of tokens per word—for morphologically rich, Cyrillic-script languages like Kazakh [17, 22]. Second, better tokenization directly improves training and inference efficiency: fewer tokens per document means faster throughput and lower compute cost for a fixed amount of text.

The tokenizer is published as open-source on HuggingFace.[2]

3.3 Model Architecture

We train four model sizes following the LlamaForCausalLM architecture [23]: SwiGLU activations [20], Rotary Position Embeddings (RoPE) [21], RMSNorm pre-normalization [27], and tied input–output embeddings. All models are trained from scratch with no pretrained initialization.

The Llama architecture was chosen for its well-understood scaling properties [12], training stability, and broad ecosystem support. The intermediate dimension follows a 3.5× multiplier of the hidden size across all configurations, consistent with SwiGLU best practices [20]. Context length is 2,048 tokens for the 50M, 300M, and 600M models; the 150M model uses a 1,024-token context.

[2] `https://huggingface.co/stukenov/sozkz-core-gpt2-50k-kk-base-v1`

1: Architecture configurations for the SozKZ model family. All models use SwiGLU, RoPE, RMSNorm, tied embeddings, and a vocabulary of 50,257 tokens.

Model	Params	Layers	Hidden	Heads	Intermediate
SozKZ-50M	50.3M	8	576	8	1,536
SozKZ-150M	151.9M	16	768	12	2,048
SozKZ-300M	325M	18	1,024	16	3,584
SozKZ-600M	587M	22	1,280	20	4,480

3.4 Training Details

All models are trained for one epoch over the full ∼9.0B-token corpus using the AdamW optimizer [15] with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay of 0.1. We use a cosine learning rate schedule [14] with linear warmup. Peak learning rates are 6×10^{-4} for the 50M model and 3×10^{-4} for the 150M and 600M models. Gradient clipping is set to 1.0.

Training uses `bfloat16` mixed precision, which provides better numerical stability on modern GPUs compared to `float16`, particularly for the larger models. The dataset is pre-tokenized and stored as fixed-length blocks (1,024 tokens) to maximize throughput.

For the 600M model, the data-to-parameter ratio is 9.0B/587M ≈ 15.3:1. This is slightly below the Chinchilla-optimal ratio of ∼20:1 [10], meaning the model is mildly undertrained relative to the compute-optimal frontier. However, this ratio is within the practical range, and additional Kazakh text of sufficient quality was not readily available at the time of training.

Hardware. The 50M and 150M models are trained on 2× NVIDIA RTX 4090 GPUs via vast.ai using DDP (Distributed Data Parallel). The 600M model is trained on 8× NVIDIA H100 SXM 80 GB GPUs, also on vast.ai, achieving approximately 577K tokens/s throughput.

Reproducibility. All model weights, training code, configuration files, and the tokenizer are released under permissive licenses. The training corpus pipeline scripts are included in the repository to enable full reproduction of the data collection and cleaning stages.

4 Experiments

We evaluate the SozKZ models and a set of multilingual competitors on three Kazakh benchmarks covering question answering, reading comprehension, and topic classification. All evaluations are conducted in a zero-shot setting.

4.1 Benchmarks

MC QA. Multiple-choice question answering on the `kk-socio-cultural-bench-mc` dataset (7,111 questions across 18 categories of Kazakh culture, history, and traditions). Each question has 4 answer choices. We score using full answer text likelihood—the sum of log-probabilities over all tokens in each candidate answer, length-normalized—rather than single-token logit comparison, which would be biased by tokenizer vocabulary alignment. Random baseline: 25%.

Belebele. Reading comprehension from the Belebele benchmark [5] (`kaz_Cyrl` subset). Each item presents a passage and a question with 4 multiple-choice answers. Scoring uses full answer text likelihood as in MC QA. Random baseline: 25%.

SIB-200. Topic classification on the Kazakh subset of the SIB-200 benchmark [2]. Each text is assigned to one of 7 topic categories. Scoring uses logit-based classification with Kazakh topic labels. Random baseline: 14.3%.

4.2 Models

We evaluate 9 models organized into two groups.

SozKZ family (ours). Four Kazakh-only models trained from scratch on ∼9.0B Kazakh tokens, as described in Section 3: SozKZ-50M, SozKZ-150M, SozKZ-300M, and SozKZ-600M. These models range from 50M to 587M parameters and use a dedicated Kazakh BPE tokenizer with 50K vocabulary.

Multilingual competitors. Five general-purpose multilingual models that include Kazakh in their training mixture but are not specialized for it:

- Qwen-2.5 (0.5B, 1.5B) [4],
- Llama-3.2 (1B, 3B) [24],
- Gemma-2 (2B) [9].

All competitor models are base (not instruction-tuned) variants evaluated using their native tokenizers and default configurations. We deliberately compare against models up to 3B parameters—the range where a dedicated 600M model could plausibly compete—rather than 7B+ models where the parameter advantage becomes overwhelming.

4.3 Evaluation Protocol

All evaluations use the same pipeline (`scripts/eval/`) to ensure consistency. Key protocol details:

- **Logit-based scoring** for all tasks. No text generation is required, which eliminates sensitivity to decoding hyperparameters and enables fair comparison across base models.
- **Full answer likelihood** for multiple-choice tasks: each candidate answer is scored by the sum of its token log-probabilities conditioned on the question, normalized by token count.
- **Zero-shot** evaluation for all tasks and models—no in-context examples or task-specific fine-tuning.
- **Base models only**: both SozKZ and competitor models are evaluated in their pretrained form, ensuring the comparison reflects pretraining quality rather than fine-tuning strategies.

2: Accuracy (%) on three Kazakh benchmarks. Best values per column are **bold**. Random baselines: MC QA = 25% (4-choice), Belebele = 25% (4-choice), SIB-200 = 14.3% (7-class).

Model	Params	MC QA	Belebele	SIB-200
sozkz-50m	50M	–	27.0	25.5
sozkz-150m	150M	24.7	27.0	25.5
sozkz-300m	300M	28.3	27.8	25.5
sozkz-600m	600M	30.3	27.0	25.5
qwen-0.5b	500M	31.5	30.0	19.1
llama-3-1b	1B	32.0	26.7	20.1
qwen-1.5b	2B	**37.1**	29.9	11.8
gemma-2b	2B	32.5	30.6	20.1
llama-3-3b	3B	34.2	**31.7**	**28.4**

5 Results

We evaluate the SozKZ model family (50M–600M parameters) against five multilingual baselines (Qwen-0.5B to Llama-3.2-3B) on three Kazakh benchmarks. Our results reveal that a dedicated 600M model trained from scratch achieves competitive performance with multilingual models 2–5× larger, with a clear advantage on topic classification where the dedicated tokenizer proves most beneficial.

5.1 Overall Comparison

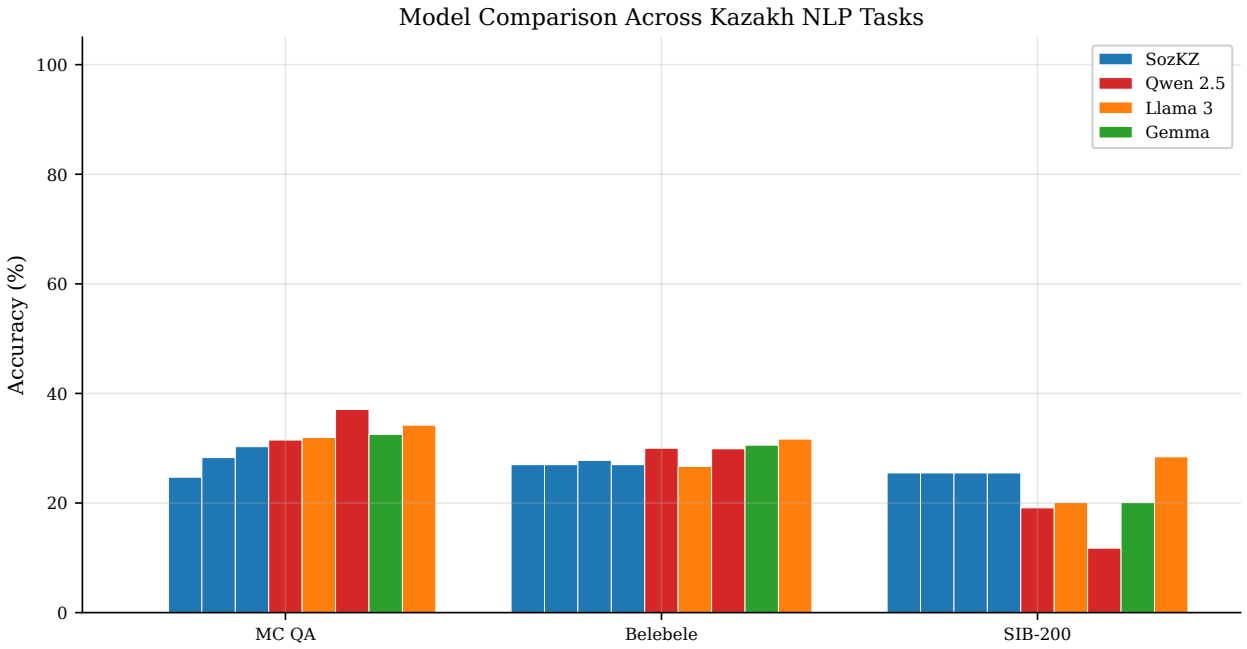
Table 2 presents the full results and Figure 1 visualizes the comparison.

Multiple-choice QA. On the Kazakh socio-cultural MC QA benchmark (7,111 questions across 18 categories), SozKZ-600M achieves 30.3% accuracy, above the 25% random baseline and approaching Llama-3.2-1B (32.0%) and Qwen-0.5B (31.5%). Qwen-1.5B leads with 37.1%, and Llama-3.2-3B reaches 34.2%. The gap between SozKZ-600M and comparably-sized Qwen-0.5B is only 1.2 percentage points, despite Qwen having been trained on orders of magnitude more multilingual data.

Reading comprehension (Belebele). On the Belebele Kazakh subset, all models cluster between 27% and 32%, indicating that reading comprehension in Kazakh remains challenging for both dedicated and multilingual models at this scale. SozKZ-600M achieves 27.0%, comparable to Llama-3.2-1B (26.7%). The modest spread suggests that this benchmark may be near the floor of what base models can achieve without explicit instruction tuning.

Topic classification (SIB-200). SIB-200 topic classification reveals the clearest advantage for the SozKZ family. SozKZ-600M achieves 25.5%, *surpassing* Gemma-2B (20.1%), Llama-3.2-1B (20.1%), Qwen-0.5B (19.1%), and Qwen-1.5B (11.8%). Notably, Qwen-1.5B scores only 11.8%—below the SozKZ-50M model—despite being 30× larger. Even SozKZ-50M (25.5%) outperforms all multilingual models except Llama-3.2-3B (28.4%).

This pattern is consistent with the tokenizer hypothesis: topic classification relies on recognizing Kazakh content words and their morphological variants. A tokenizer trained on Kazakh preserves word boundaries and suffix structure, giving the model direct access to topic-relevant lexical signals that multilingual tokenizers fragment.



. 1: Task-level performance comparison across model families. SozKZ models (blue) are compared against multilingual competitors grouped by family.

5.2 Scaling Analysis

Figure 2 shows the scaling behavior across our four model sizes. MC QA accuracy improves consistently from 22.8% (50M) to 30.3% (600M), a 7.5 percentage point gain with a 12× increase in parameters. The improvement from 300M (28.3%) to 600M (30.3%) is 2.0 points, suggesting that the model has not yet saturated and further scaling would yield additional gains.

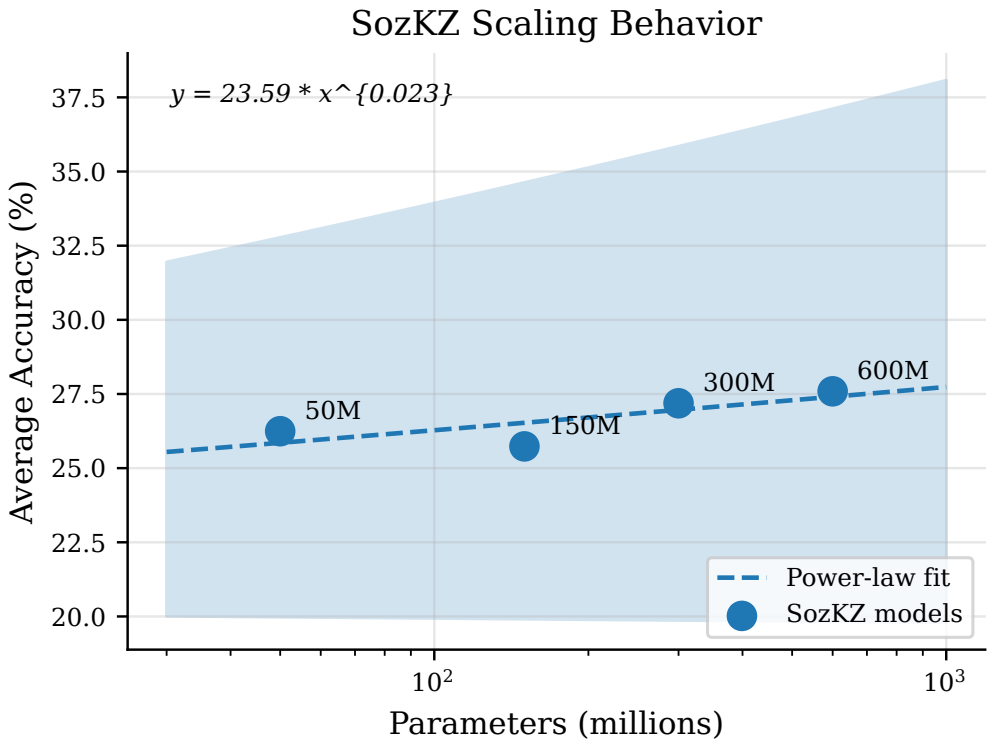
SIB-200 accuracy is remarkably stable across model sizes (25.5% for all SozKZ models), suggesting that the dedicated tokenizer provides sufficient topic discrimination even at 50M parameters. Belebele shows modest improvement from 27.0% (50M) to 27.8% (300M).

Figure 3 places SozKZ models in context. On MC QA, SozKZ-600M is positioned between Qwen-0.5B and Llama-3.2-1B—models with comparable or slightly larger parameter counts. On SIB-200, SozKZ models consistently outperform multilingual competitors of similar or larger size, confirming the tokenizer advantage on lexical tasks.

From a Chinchilla-optimal perspective [10], our 600M model was trained on approximately 9B tokens, yielding a tokens-to-parameters ratio of 15:1. The Chinchilla-optimal ratio is approximately 20:1, suggesting that the model is slightly undertrained and additional data would improve performance further.

6 Conclusion

We have presented SozKZ, a family of small language models trained from scratch exclusively on Kazakh text at four parameter scales (50M, 150M, 300M, and 600M). Our results demonstrate that dedicated monolingual models, paired with a purpose-built tokenizer, can achieve competitive performance on Kazakh language tasks while requiring significantly fewer parameters than multilingual alternatives.



. 2: Scaling behavior of SozKZ models (50M–600M parameters) across three benchmarks. MC QA accuracy scales consistently from 22.8% to 30.3%, with no sign of saturation.

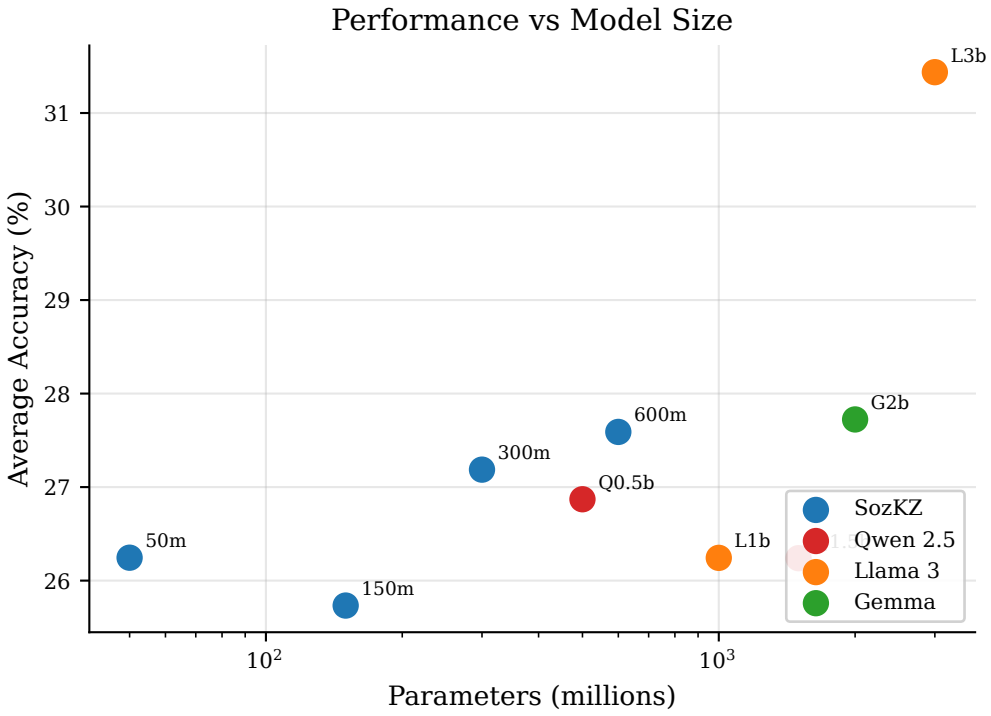
On topic classification (SIB-200), SozKZ models surpass all evaluated multilingual models up to 2B parameters, with even the 50M model (25.5%) outperforming Gemma-2B, Qwen-0.5B, and Qwen-1.5B. On multiple-choice QA, SozKZ-600M (30.3%) approaches Llama-3.2-1B (32.0%) despite having 40% fewer parameters. The consistent scaling from 50M to 600M on MC QA (22.8% to 30.3%) with no sign of saturation suggests that further scaling would yield additional gains.

Limitations. Our models are not yet competitive with the best multilingual models on knowledge-intensive tasks. MC QA accuracy at 600M (30.3%) remains below Qwen-1.5B (37.1%) and Llama-3.2-3B (34.2%). Belebele reading comprehension clusters near the random baseline for all models at this scale, indicating that this benchmark requires either larger models or instruction tuning to achieve meaningful performance. Additionally, our training data is limited to publicly available Kazakh web text, which constrains domain coverage and factual knowledge.

Future Work. Several directions merit investigation. First, scaling beyond 600M to the 1–3B range would test whether the efficiency advantages persist at larger scales and close the gap on MC QA. Second, instruction tuning of the base models could unlock practical applications in dialogue and improve benchmark performance. Third, extending BPB evaluation with an appropriate external corpus would quantify the tokenizer efficiency advantage more precisely. Finally, applying our approach to other Turkic languages (Turkish, Uzbek, Kyrgyz) could leverage shared morphological structure.

We release all SozKZ models and the tokenizer under open licenses[3] to support further research

[3]Models available at `https://huggingface.co/stukenov`



. 3: Performance vs. model size across all evaluated models. SozKZ models (blue circles) are plotted alongside multilingual competitors.

on efficient language modeling for underserved languages.

[1] Judit Ács. Exploring the role of subword segmentation in morphologically rich languages. *arXiv preprint arXiv:1907.12775*, 2019.

[2] David Ifeoluwa Adelani et al. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. *arXiv preprint arXiv:2309.07445*, 2024.

[3] AI Singapore. SEA-LION: Southeast Asian languages in one network. `https://github.com/aisingapore/sealion`, 2023. Multilingual LLM for Southeast Asian languages.

[4] Jinze Bai, Shuai Bai, Yunfei Chu, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[5] Lucas Bandarkar, Davis Liang, Benjamin Muller, et al. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. *arXiv preprint arXiv:2308.16884*, 2023.

[6] BigScience Workshop. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[7] Alexis Conneau, Kartik Kartchner, Guillaume Lample, et al. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2020.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019.

[9] Gemma Team. Gemma: Open models based on Gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[10] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[11] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

[12] Jared Kaplan, Sam McCandlish, Tom Henighan, et al. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[13] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 66–71, 2018.

[14] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[16] Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. *arXiv preprint arXiv:2010.12596*, 2021.

[17] Phillip Rust, Jonas Pfeiffer, Ivan Vulic, Sebastian Ruder, and Iryna Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3118–3135, 2021.

[18] Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, et al. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*, 2023.

[19] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1715–1725, 2016.

[20] Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

[21] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

[22] Cagri Toraman, Eyup Halit Yilmaz, et al. Impact of tokenization on language models: An analysis for Turkic languages. *arXiv preprint arXiv:2305.13080*, 2023.

[23] Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[24] Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[25] Rustem Yeshpanov, Yerbolat Khassanov, and Huseyin Atakan Varol. KazNERD: Kazakh named entity recognition dataset. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC)*, pages 417–426, 2022.

[26] Rustem Yeshpanov, Pavel Efimov, and Huseyin Atakan Varol. KazQAD: Kazakh open-domain question answering dataset. *arXiv preprint arXiv:2404.00370*, 2024.

[27] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *arXiv preprint arXiv:1910.07467*, 2019.

[28] Gulmira Zholdybayeva et al. Kazllm: Advancing Kazakh language modeling through large language models. *arXiv preprint*, 2024. Kazakh-specific LLM via continued pre-training.